The Role of Field Data for Analyzing the Dependability of Short Range Wireless Technologies

G. Carrozza[1] and M. Cinque[1]

Dipartimento di Informatica e Sistemistica, Università di Napoli Federico II
Via Claudio 21, 80125 - Napoli, Italy
{ga.carrozza, macinque}@unina.it

Abstract. *The migration from mobile to ubiquitous Internet is at hand, due to the intense growth of short range wireless technologies. Users accessing the Internet through wireless devices are increasing, if compared to "wired" ones, and they expect the same dependability level they already experience on wired networks, that is high quality "always on" wireless networks. But how can we analyze the dependability level of a wireless network? Direct analysis of failures from the field of application is an effective practice to understand the actual dependability behavior of an operational system. However, despite its wide use over the last four decades on a large variety of systems, field data analysis has rarely been applied to wireless networks. Through the experience gained from extensive failure analysis of Bluetooth networks, the article shows how field failure data can play a key role to fill the gap on understanding the dependability behavior of wireless networks.*

1 Introduction

Long time has passed since Meyer proposed of the idea of "Ubiquitous Computing", the paradigm which aims at *enhancing computer use by making many computers available throughout the physical environment*, and at making computers *effectively invisible to the user* [12]. Since then, embedded systems engineering and wireless communications have progressed fast, thus making the visionary idea of *Ubiquitous Computing* a reality. The intense device miniaturization and the increasing power of microprocessors, along with the availability of cheap wireless networks and connectivity, allows computers to increasingly pervade everyday human life and activities.
Longer time has even passed since the Internet was anchored to telephone wires and coaxial cables. Since 2005, cell phones have outnumbered PCs and, in the last few years, people access the Internet more from a wireless device than from a wired one, thus enabling mobile Internet access. According to ITU reports [1], mobiles dominate both in quantity and in quality. Small embedded devices have become a daily portable necessity, which is always no more than one meter away

[1] International Telecommunication Union, www.itu.int/osg/spu/presentations

from users. PDAs, laptops, cellphones, MP3 players, webcams, and even fridges and microwave ovens, have embedded Internet connectivity, and allow to access the global network from everywhere. The power of mobile has cut off geographic boundaries, hence networks, people and devices are seamlessly connected both on a local scale and over the world: the transition from the Internet to the Ubiquitous Internet is at hand.

Short Range Wireless (SRW) technologies are at the core of such a revolution, as well as the key to ubiquitous networking. They are primarily meant for indoor use and over short ranges, in which they are able to connect portable devices with high connection speed and low power consumption. They are often used at the edges of the wired network, e.g., as wire replacement, to provide mobile users with the last hop to the Internet, from anywhere and at anytime.

Nevertheless, higher mobility means lower speed, as well as worst connection quality in terms of transmission capacity and reliability. Hence, many technical challenges have to be faced in order to serve today customers' demand, who expect the same level of quality they already experience on wired networks. In addition, the wide range of business critical applications in which SRW technologies are protagonists (e.g., mobile banking, mobile commerce, etc), along with their usage in mission critical scenarios (e.g., remote control of robots, rescue of catastrophe survivors, etc.) make it crucial to answer a simple question: *can we rely on these technologies*?

This simple question has not a simple answer. Research efforts in the field of dependability, wireless networks and ubiquitous systems, have to be merged to give a satisfactory response. Indeed, a non-negligible knowledge of the dependability behavior of SRW technologies is required in terms of what are the failure modes, how can we describe/model them, what are the dependability pitfalls and consequences to applications, and how can we face them.

Field Failure Data Analysis (FFDA) is an effective mean to gain the required knowledge. It consists in observing spontaneous occurrences of failures of an operational system, without forcing or inducing artificial failures in the system. The collected failure data provide accurate information which can characterize the dependability of the system under study.

FFDA has been successfully applied in the last four decades. Several studies have been conducted on a large variety of systems, including operating systems and the Internet. As for the former, hangs and the well known "blue screens", found on Windows NT 4 to be mostly due to application failures, were significantly reduced in the successive generation of the OS, Windows 2000, providing the kernel with greater isolation from errant applications [9]. As for the latter, [10] analyzes the causes of failures and the potential effectiveness of various techniques for preventing and mitigating failures in large-scale Internet services.

Despite the large use in both the academy and the industry, FFDA has rarely been used to characterize the dependability of wireless access networks. In this article we aim to show how field failure data can play a key role to gain the needed knowledge to model the failure behavior and to uncover dependability pitfalls of wireless access networks. The resulting understanding is essential for

the effective design of any new solution for dependable wireless networking.
We focus on the Bluetooth technology, which has lots of potential applicability in the "last meter" for personal area networks (PANs). It has been estimated that in 2005 Bluetooth was a built-in feature for more than 600 million products, manufactured by several companies. CSR (Cambridge Silicon Radio), in its 2007 financial report, said it expects the proportion of new cars that include Bluetooth to increase from 5 up to 30 percent in the medium term. Car-kits use GPS high performance solutions embedded into a Bluetooth chip, thus bringing GPS into a wide range of new low-cost devices. Furthermore, portable devices are being more and more equipped with both Bluetooth and IEEE 802.11 (Wi-Fi), hence Bluetooth represents a cost-effective way to improve the connection availability in the case Wi-Fi networks are not available; as the number of Access Points to the Internet increases, Bluetooth demonstrated to scale better than Wi-Fi in terms of bandwidth, delay, fairness and energy efficiency [5].
This article provides an answer to the fundamental question posed above in the context of Bluetooth networks, by exploiting over four years authors' research experience on FFDA of mobile/wireless environments [3, 2, 8]. Conducted experiments allowed to define and to statistically model the failure modes of Bluetooth according to the layer they occur, i.e., application, system (Bluetooth stack and operating system), or wireless channel layer, according to both a *user-centric* and a *channel-centric* approach (see Section 2). Some of the key findings are summarized in the following. First, severe failures, such as connection failures and packet losses, may manifest to applications every eight minutes, on average. This is partially due to the bursty nature of observed channel failures, which are more likely to elude integrity checks performed by Bluetooth, hence propagating to the operating system and applications. Second, failures revealed in the absence of Wi-Fi interferences are rarer, but more severe and harder to recover than when Wi-Fi is present. Third, Bluetooth transport layers assume underlying data-link layers to be completely reliable, hence they do not perform error and integrity checks. However, presented results show that these layers are not able to tolerate low level failures.
These findings provide valuable insights that have to be considered when designing Bluetooth-based access networks with demanding dependability and ubiquity requirements.

2 A Combined Perspective to Gain From Field Data

FFDA studies usually account three consecutive steps: i) data logging and collection, where data are gathered from the operational system, usually exploiting system log files or failure reports, ii) data filtering and manipulation, concerning the extraction of the information which is useful for the analysis, and iii) data analysis, i.e., the derivation of the intended results from the manipulated data.
The operational system can be observed according to both a *top-down* and a *bottom-up* approach. The former is a well known practice in the field of dependability evaluation and measurement [10, 4, 3] that allows to infer the fail-

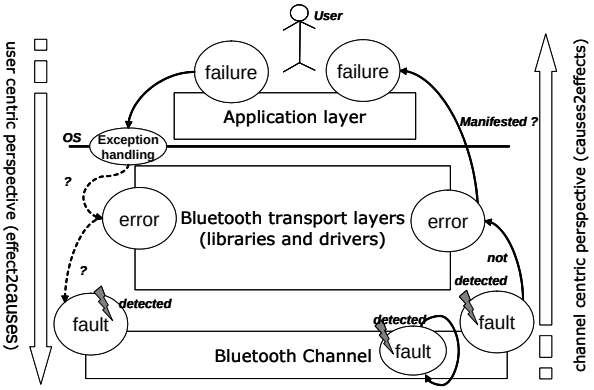


Fig. 1. User- and Channel-centric perspectives

ure causes starting from the effects on application and Operating System (OS) layers, according to users' point of view (*user-centric* perspective). The latter, instead, is based on the novel idea of tracing how faults propagate to upper layers by directly observing low level causes [2]. With respect to wireless systems, this is a *channel-centric* perspective, in that data communication channel is the starting point for system observation.

Figure 1 emphasizes the differences between the two approaches, with reference to the Bluetooth stack. It is important to precise that, according to the terminology introduced in [1], channel failures can be seen as errors for system failures and as faults for application failures, as well as system failures can be seen as errors for application failures. The user-centric approach allows to analyze failure propagation traces only down to the OS level (Bluetooth drivers failures are logged on system log files). Conversely, by adopting a channel-centric approach, it is possible to monitor failures occurring at the Bluetooth data-link layer, namely Baseband, and to evaluate its *coverage* (i.e., Baseband's ability of self repairing corrupted frames).

In this work we show how combining both the perspectives provides a detailed characterization of SRW technologies dependability. In particular, by following this approach we are able to classify Bluetooth failure modes according to the layer they manifest, and to gain insights into failure propagation traces.

2.1 Testbed and Workload

Field data have been collected by running experiments on a real-world Bluetooth piconet (i.e. a network made up of 1 to 8 Bluetooth nodes, only one of them acting as the *master* or coordinator). Bluetooth piconets can be easily exploited to access the Internet, by means of the Bluetooth Personal Area Network (PAN) profile. A user willing to surf the web with his Bluetooth-enabled mobile phone, starts an inquiry/scan operation to discover other devices in the neighborhood,

then - through a Service Discovery Protocol (SDP) operation - he looks for the Network Access Point (NAP). Once the NAP has been found, the user connects to it (note that the connection operation usually takes care of switching the role of the mobile device to slave, letting the NAP be the master of the piconet). Finally, the user can happily navigate to his web-mail inbox.
An application workload (WL) has been designed to emulate the behavior of a typical PAN user. The WL performs all the steps needed to setup the PAN, as mentioned above. The WL then stimulates the wireless channel by transferring data on it. To add uncertainty to piconet evolution, each WL cycle is characterized by several random variables modeling both connection establishment (e.g. whether the inquiry/scan and SDP procedures are performed or not) and channel usage (e.g. according to the random variables which are used to model actual Internet traffic, such as Web surfing, file transfer, e-mail, etc.). Running the WL, and collecting both application and system failures registered on OS log files is useful to achieve the user-centric perspective. During packets transmission, channel level data have been captured by using a Bluetooth air sniffer, in order to achieve the *channel-centric* perspective. The sniffer provided us with all the needed information, from failure reports at the Baseband layer to frame status as they are delivered up to L2CAP (Logical Link Control and Access Protocol, i.e., the Bluetooth transport) and BNEP (Bluetooth Network Encapsulation Protocol, which is used to emulate Ethernet links over Bluetooth).
The produced failure data come from multiple sources (WL log files, system log files, and sniffer traces). Combining these data with temporal coalescence algorithms permits to infer propagation traces from channel up to the application layer. Data have been properly filtered to discard useless information.
Several experiments have been conducted on the piconet, during a time span of almost two years, collecting more than 140 millions failure data items. In order to investigate the impact of Wi-Fi on Bluetooth failure modes, they have been performed both in presence and in absence of Wi-Fi disturbances.

3 Bluetooth Failure Modes

Field failure data demonstrate to be an effective mean to identify the failure modes of SRW technologies.
In our case, we were able to observe several failure modes and to classify them according to the level in which their occurrence is registered. Observed Failure modes are summarized in Fig.2. Applications exhibit a variety of failures according to the utilization phase where they occur, i.e., inquiry/scan and discovery phases, PAN connection, and data transferring. Failures during the connection can occur either while the connection is set up or while the role of the device is switched from master to slave. Unexpectedly, failures during data transfer, such as packet loss and mismatches in the received data, are experienced, despite error control mechanisms performed by Baseband, such as Cyclic Redundancy Codes (CRCs), Forward Error Correction (FEC), and Header Error Correction (HEC) schemes. However, as discussed in [6], the weakness of integrity checks is the

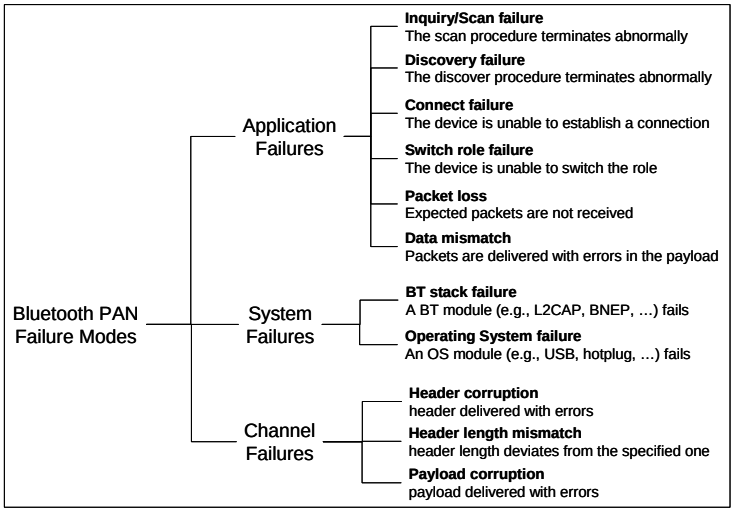


Fig. 2. Bluetooth Failure Modes as they are observed on the field

assumption of having memoryless channels with uncorrelated errors from bit to bit. In the case of Bluetooth, correlated errors (e.g. bursts) can occur due to the nature of the wireless media, affected by multi-path fading and electromagnetic interferences. The failure of the integrity checks is further investigated in the next section.

System level failures are grouped with respect to their location, i.e., Bluetooth software stack and Operating System. Failure types could be further refined according to the component which signals the failure, e.g., L2CAP and BNEP.

Finally, three channel level failures classes have been identified: (i) header corruption at the Baseband level, (ii) length mismatch, i.e., a mismatch in the packet length reported into the Baseband header and the actual one, and (iii) payload corruption (PC in the following) at the Baseband level. The last class deserves more attention, since these failures are the only ones that may propagate to system and application layers, in our settings. Failures belonging to the other two classes are instead successfully detected and masked by the Baseband's FEC, HEC and CRC schemes.

Failure data also allow to model failure dynamics as stochastic processes. The statistical distribution type then permits to better understand the failure phenomenology. In our case, we attempted to fit the time to failure (TTF) for application failures with three different statistical distributions: the Exponential, the Lognormal, and the Weibull distributions. The fitting has been conducted by means of a statistical software suite, using maximum likelihood estimators and goodness of fit tests. It results that almost all application level failures are distributed as Lognormal. The Lognormal distribution is used extensively in reliability applications to model failure times. A random variable can be modeled as Lognormal if it can be thought of as the multiplicative product of many small

independent factors. In our case, this means that application level failures are the product of many small faults at a lower level. These faults can be both software faults, e.g., heisenbugs (i.e., design faults which conditions of activation occur rarely or are not easily reproducible [11]) at the various level of the Bluetooth stack, and channel faults, as the payload corruption case. Interestingly, only data mismatch failures are distributed as Exponential. This is coherent with the fact that, as will be observed in next section, direct cause for data mismatches are payload corruptions, which also resulted to be exponentially distributed.
More detailed analysis allows to derive interesting characteristics of the failure behavior. For instance, we attempted to characterize BT connections survivability, i.e., the duration of BT connections before they are unexpectedly lost due to failure (and not due to normal connection closing operation). We observed that the connection duration with respect to failures is statistically *self-similar*, i.e., it shows the same statistical properties at many different scales. On a side, this implies that connection duration times can be modeled with heavy tailored distributions (e.g., the Pareto distribution). On the other side, this shows evidence that connection durations exhibit *long range dependence*: the failure of a connection at a given time is typically correlated with connection failures at all future instants.

4 Bluetooth Uncovered Dependability Pitfalls

4.1 Impairments due to Payload Corruptions Propagation

As stated in section 3, there exists a class of channel level failures, namely PC, that is able to elude Baseband error control mechanisms, and to propagate to upper layers with a non zero probability. When dealing with digital wireless communication, the causes of such failures lie into shadowing and electromagnetic noise, which may cause the bits to be flipped when transfered between two end points. Moreover, as previously mentioned, the presence of multi-path fading and electromagnetic interferences can cause correlated faults (i.e. bursts) to occur.
Thanks to field experiments, and to a thorough inspection of packets content, we were able both to observe the occurrence of PC on monitored Bluetooth channel, and even to pinpoint the flipped bits.
A snapshot of a corrupted payload is shown in Fig. 3. Note that we were able to uncover this corruption since we forced the WL to transfer a known character sequence with a fixed length, e.g. "CCCC". The highlighted burst is 136 bits long. This is the reason why it is able to elude Baseband error control mechanisms. Baseband adopts a 16-bit CRC-CCITT polynomial code which is able to detect 18 bits or longer bursts with 0.99998 probability (i.e. minor than one). We experienced that the length of the burst is a random variable, L, with an expected value equals to 512 bits and a standard deviation equals to 646 bits, hence they are longer on average than 18 bits.
Graph in Fig. 4 shows how a PC can propagate. On the leftmost side of the graph, it is shown that 99.59 % of PC are detected by Baseband, hence they do not

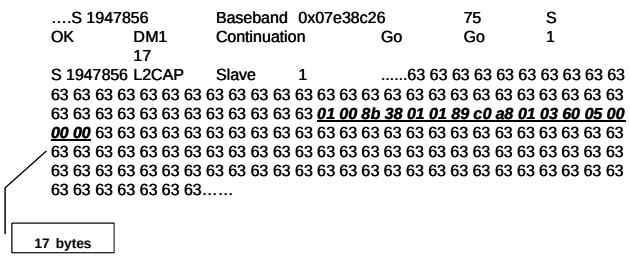


Fig. 3. Example of corrupted payload

reach upper layers. With respect to the undetected failures, values on the graph links represent the conditional probability of failures given that a PC occurred and eluded Baseband control. Several consequences can then occur, according to the probabilities reported on the graph. In fact, PC can either remain latent (i.e. isolated) at the system level, or propagate to the user level in the form of *application failures*. In the former case, they are *confined* at system level even if no further error controls are performed (in fact, both L2CAP and BNEP assume underlying levels to be completely reliable). The actual induced failure depends

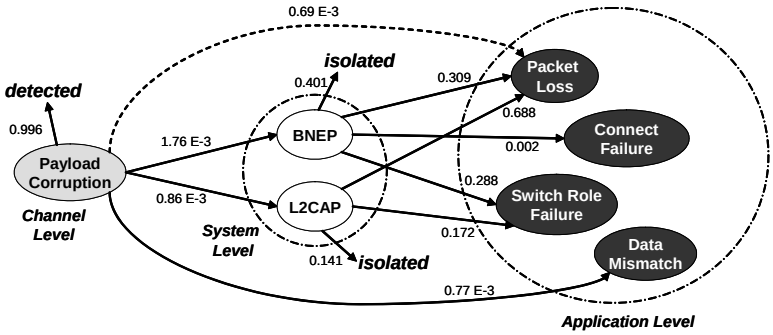
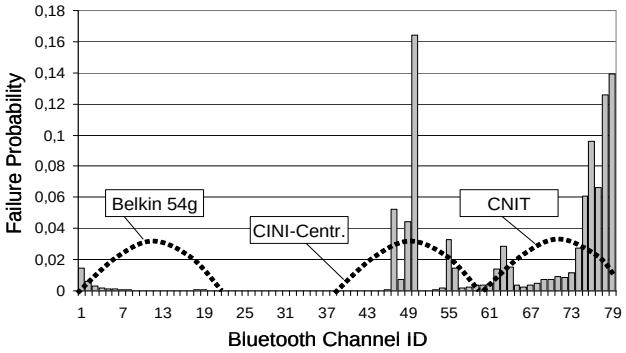


Fig. 4. Propagation phenomenology

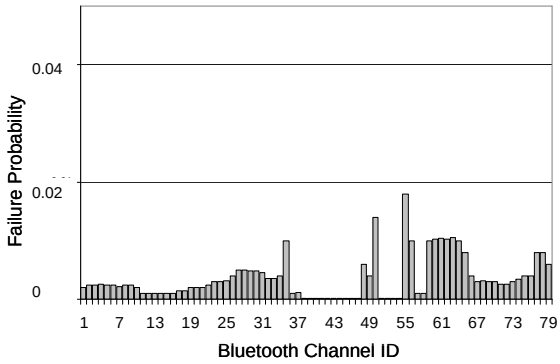
on the location of the burst within the transmitted packet. For instance, if the corruption affects the L2CAP header, the packet can not be properly decoded. As a consequence, it will not be delivered to upper layers, thus causing a packet loss, i.e. an omission failure, at the user level. In the same way, if the burst is located in the L2CAP payload, the erroneous content can be directly delivered to the application, which may then exhibit a value failure, i.e. a data mismatch in the Figure.

4.2 Problems due to the impact of Wi-Fi on Bluetooth

Many efforts have been devoted to investigate coexistence issues between Wi-Fi and Bluetooth [7]. We tried to estimate how the presence of a Wi-Fi network in the neighborhood can impact Buetooth failure modes. To this aim we let WL run both in the presence and in absence of Wi-Fi interferences.
We compared the conducted experiments in terms of Baseband failure rate and failure distribution over channels.
In the presence of Wi-Fi interferences, the Baseband failure rate has been measured as 6.822 faults per second. Since the average number of transmitted frames per second is 596, this results into a frame error rate of about 0.012 (i.e., about 1 frame out of 100). However, the most of these errors are promptly detected and masked by Baseband's correction mechanisms, in that its coverage, with respect to all channel failures, has been measured as 0.9996. Undetected failures can be modeled as an exponential random variable with a 458716 ms mean. This means that about every eight minutes a Baseband error is not detected, and a wrong frame propagates to upper layers. As one could expect, a lower failure rate (equals to 0.516 faults per second) has been experienced when Wi-Fi Access Points (APs) in the neighborhood are turned off.
As for failure distribution across wireless channels, results are shown in Fig. 5. In particular, Fig. 5(a) shows failure probability for all failures (even detected ones) over channels when WiFi is present whereas results in Fig. 5(b) refer to the experiment conducted without WiFi disturbances. In the first case, error probability is highly concentrated over the channels evidenced by dotted lines corresponding to the actual channel overlap between the three Wi-Fi APs deployed in our laboratory and the Bluetooth channels (Bluetooth uses 79 wireless channels, each 1-MHz-wide, in the unlicensed 2.4 GHz band; Wi-Fi uses eleven 22-MHz-wide sub-channels across the same band of Bluetooth; when a Bluetooth and a WiFi radio are in the same area, a single Wi-Fi channel overlaps with 22 of the 79 Bluetooth channels.). Fault probabilities strongly depend on APs usage. For instance, the AP working on channels from 1 to 23 is rarely used, thus justifying the low fault probability over these channels. Figure 5(b) shows that the probability over interfering channels drastically decreases when WiFi is absent. This is a further confirmation of the lower fault rate we measured in the absence of interferences. Interestingly, we found that faults that occurred in absence of Wi-Fi interferences were more "severe" than those that occurred when Wi-Fi is present. This conclusion can be drawn by investigating time to failure statistics for all failures (detected and undetected). In both cases, they fit a Lognormal distribution, but with different values of distribution parameters (e.g. distribution shape). This leads us to observe that in absence of Wi-Fi short inter-arrival times of failures are more probable. In other terms, the absence of disturbances causes the faults to be more clustered in time. The reason for this is to be found into the frequency hopping scheme adopted by Bluetooth. In the presence of Wi-Fi, faults are mainly due to interferences which tend to be polarized on the overlapped channels. After the occurrence of a failure due to collision, the frame is retransmitted over a different channel. However, the channel might



(a) WiFi present (overlapping zones for each AP are shown)



(b) WiFi not present

Fig. 5. Histogram of failure probability across Bluetooth channels.

either be free or still occupied by the Wi-Fi interference. This variability causes both short- and medium-length inter-failure times. When Wi-Fi is not present, there are no polarized interferences, or, in other terms, the fault phenomena is spread (e.g., lost of synchronization among nodes or wide-band disturbances). Hence, it is more likely that a retransmission will fail.

In order to corroborate this intuition, we also investigate Mean Time To Recover (MTTR) in both circumstances. Consistently with above results, MTTR increases when Wi-Fi is not present (from $7.51ms$ to $9.52ms$) , i.e. more retransmissions are needed when the fault phenomenon is not polarized. Finally, the Baseband level exhibited a lower capability of detecting failures due to spread phenomena in that its coverage decreases by one order of magnitude (it passes from 0.9996 to 0.9968). This means that failures due to spread phenomena are more prone to elude Baseband's CRC integrity check.

5 Conclusions

Short range wireless technologies are the key of ubiquitous networking. They represent the principal medium to access the Internet from mobile devices. As these technologies are widely used in business and mission critical applications, characterizing their dependability represents a significant issue. Field Failure Data Analysis shows to be an effective instrument to build up the needed knowledge on the dependability behavior of actual wireless networks. The case of Bluetooth, analyzed in the article, gives evidence of how field data help to uncover dependability pitfalls. The achieved results are useful to define mitigation actions to improve the overall dependability level of Bluetooth networks, as shown in our previous work. The same analysis need to be conducted on other wireless networks enabling ubiquity both over long distances, e.g., WiMAX, (Worldwide Interoperability for Microwave Access), and within short ranges, e.g., UWB (Ultra Wide Band), and WUSB (Wireless USB), with the aim of building large and publicly available field failure data repositories. These can be exploited by researchers and practitioners to design dependable wireless solutions.

References

1. A. Avizienis, J.C. Laprie, B. Randell, and C. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, 2004.
2. G. Carrozza, M. Cinque, D. Cotroneo, and S. Russo. Dependability Evaluation and Modeling of the Bluetooth Data Communication Channel. *Proc. of the 16th Euromicro International Conference on Parallel, Distributed and network-based Processing*, February 2008.
3. M. Cinque, D. Cotroneo, and S. Russo. Collecting and Analyzing Failure Data of Bluetooth Personal Area Networks. *Proc. of the 36th IEEE International Conference on Dependable Systems and Networks (DSN'06)*, June 2006.
4. R. K. Iyer, Z. Kalbarczyk, and M. Kalyanakrishnam. Measurement-Based Analysis of Networked System Availability. *Performance Evaluation Origins and Directions, Ed. G. Haring, Ch. Lindemann, M. Reiser, Lecture Notes in Computer Science 1769, Springer Verlag*, 2000.
5. P. Johansson, R. Kapoor, M. Kazantzidis, and M. Gerla. Personal Area Networks: Bluetooth or IEEE 802.11? *International Journal of Wireless Information Networks Special Issue on Mobile Ad Hoc Networks*, April 2002.
6. P. Koopman and T. Chakravarty. Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks. *Proc. of the 34th IEEE International Conference on Dependable Systems and Networks (DSN '04)*, June 2004.
7. J. Lansford, A. Stephens, and R. Nevo. Wi-Fi (802.11b) and Bluetooth: Enabling coexistence. *IEEE Network*, pages 20 – 27, September/October 2001.
8. Z. Kalbarczyk R. K. Iyer M. Cinque, D. Cotroneo. How do mobile phones fail? a failure data analysis of symbian os smart phones. In *Proc. of the 37th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2007), Edinburgh 2007*, pages 585–594, 2007.
9. B. Murphy and B. Levidow. Windows 2000 Dependability. MSR-TR-2000-56, Microsoft Research, Microsoft Corporation, Redmond, WA, June 2000.

10. D. Oppenheimer, A. Ganapathi, and D.A. Patterson. Why do Internet services fail, and what can be done about it? *Proc. of the 4th USENIX Symposium on Internet Technologies and Systems (USITS '03)*, March 2003.
11. K. Vaidyanathan and K. S. Trivedi. A Comprehensive Model of Software Rejuvenation. *IEEE Transactions on Dependable and Secure Computing*, 2(2):124–137, April-June 2005.
12. Mark Weiser. Ubiquitous computing. *IEEE Computer*, 26(10), October 1993.